FakeX: A Framework for Detecting Fake Reviews of Browser Extensions

Eric Olsson
Chalmers University of Technology
Gothenburg, Sweden

Benjamin Eriksson
Chalmers University of Technology
Gothenburg, Sweden

Pablo Picazo-Sanchez
School of Information Technology,
Halmstad University
Halmstad, Sweden
Chalmers University of Technology
Gothenburg, Sweden

Lukas Andersson
Chalmers University of Technology
Gothenburg, Sweden

Andrei Sabelfeld
Chalmers University of Technology
Gothenburg, Sweden

ABSTRACT

Browser extensions boost user experience on the web. Similarly to smartphone app stores, browsers like Chrome distribute browser extensions via their Web Store, enabling a thriving market of third-party developed extensions. The Web Store incorporates a user review system to help users decide which extensions to install. Unfortunately, the open nature of the review system is subject to reputation manipulation. As browser vendors fight reputation manipulation, attackers employ more sophisticated methods to stay under the radar. Focusing on fake reviews, we identify several techniques attackers use: fake accounts, disjoint sets of fake accounts for different extensions, automation of generated reviews, and focusing on reviews rather than ratings. We present FakeX, a framework to detect fake reviews by focusing on inference from review metadata. FakeX employs five distinct methods, including temporal distribution analysis, relationship clustering, and ratio-based assessments, to unveil patterns indicative of fake reviews. Evaluation of over 1.7 million reviews reveals the effectiveness of FakeX in identifying hundreds of fake review campaigns. Furthermore, our investigation of these fake reviews uncovers 86 malicious extensions, mounting attacks that range from data-stealing to monetization, impacting over 64 million users. In addition, we collaborate with Adblock Plus and Avast to demonstrate FakeX in action, expanding a seed list of newly detected malicious extensions to discover a further 16 malicious extensions with millions of users, where, in some cases, attackers tried to improve malicious code.

CCS CONCEPTS

• **Security and privacy** → **Browser security**; **Web application security**.

KEYWORDS

Browser Extensions, Fake Reviews, Web Security

ASIA CCS '24, July 1–5, 2024, Singapore, Singapore
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0482-6/24/07.
https://doi.org/10.1145/3634737.3656999

ACM Reference Format:
Eric Olsson, Benjamin Eriksson, Pablo Picazo-Sanchez, Lukas Andersson, and Andrei Sabelfeld. 2024. FakeX: A Framework for Detecting Fake Reviews of Browser Extensions. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '24), July 1–5, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3634737.3656999

1 INTRODUCTION

Browser extensions are user-friendly applications that personalize the browsing experience by introducing features and/or modifying the appearance of web pages. Developed using standard web languages like HTML and JavaScript, extensions empower developers to easily craft applications that interact smoothly with the browser's components and user interface. These extensions have attracted millions of users globally [7], contributing to the popularity of extension-enabled browsers such as Google Chrome.

Web browser vendors feature app stores for distributing approved extensions. The foremost example is the Chrome Web Store, which distributes extensions for the Chrome web browser as well as for other Chromium-based browsers such as Brave and Opera.

The Web Store incorporates a user review system, allowing users to share feedback on their installed extensions—uniquely identified by an ID[1]. This system is crucial for promoting high-quality extensions and assisting users in deciding which extensions to install [21]. Users can rate extensions on a 1 to 5-star scale and provide written reviews.

Unfortunately, the open nature of the Web Store's review system has led to the emergence of reputation manipulation. Reputation manipulation occurs when individuals or groups artificially enhance or undermine an extension's reputation, often through fake reviews and ratings. The problem is exacerbated by security experts recommending that users read reviews to enhance their Internet safety [6, 9, 38, 43]. This issue impacts the platform's credibility and may cause users to install low-quality or malicious extensions [45].

Faking reviews has become an attractive target for monetization on the black market [31], with several websites and communities offering to sell reviews online [15, 16, 42, 50–52]. These activities

[1] For example, the official Google Translate extension has the unique ID aapbdbdomjkkjkaonfhkkikfgjllcleb. All IDs used in this study are included in Appendix E.

thrive despite vendors like Google explicitly forbidding reputation manipulation [19], including attempts at inflating reviews and ratings [18].

At the same time, detecting fake reviews is challenging, which explains why they persist on stores like the Web Store [27]. As we will see, modern fake reviews attempt to stay under the radar by avoiding obvious faking techniques so they will not be flagged for anomalies, such as excessive reviews from a single user. Motivated by these challenges, we propose two research questions:

RQ1: Can fake reviews be detected on the Chrome Web Store?
RQ2: Can methods for detecting fake reviews help discover malicious extensions?

The text of fake reviews is often generic, making it hard to distinguish from benign reviews. Sometimes, there is even no distinction as vague text can be copied from legitimate reviews to reapply as fake reviews to a broad swath of extensions. Furthermore, fake review authors employ various concealment techniques to stay undetected. Indeed, we identify several techniques fake review authors use to operate while staying under the radar: fake accounts, disjoint sets of fake accounts for different extensions, automation of generated reviews, and focusing on reviews rather than ratings. A key observation is that we can still track these techniques through the temporal metadata of the reviews and user IDs.

To investigate our research questions, we introduce FakeX, a framework for detecting fake reviews. A principal strength of FakeX is that it does not rely on the reviews' content but focuses on the metadata, which includes the timestamps associated with users' Web Store reviews. In particular, we focus on 1) the temporal distribution of reviews, 2) the relationship between reviewers, and; 3) the ratios between ratings and reviews.

FakeX comprises five main methods, where three focus on the temporal distribution of the reviews, whereas the other two use the relationships between reviewers. In particular, FakeX offers 1) Aggregated Time Window (ATW), a novel method for identifying multiple accounts who post reviews in close temporal proximity; 2) Horizontal Vertical Clustering (HVC), a machine learning-based approach that clusters reviews by their timestamps, not only within the same extension but also across multiple extensions; 3) Spam Detection, a method focused on extracting bursts of high review activity in a short period within an extension; 4) Co-Reviewer (CoR), a method that focuses on discovering clusters of reviewers who consistently review the same extensions, and; 5) Written Ratio, a method that use the ratio between rating and reviews to find extensions with an exceptionally high fraction of written reviews. Table 1 summarizes the methods used to create fake reviews and which of our methods detect them.

To evaluate FakeX, we download all 1,782,702 reviews of all 115,124 extensions in the Web Store as of February 9, 2023. Answering RQ1 positively, FakeX uncovers hundreds of review campaigns sharing large numbers of reviewers, some consisting of thousands of accounts. One method in FakeX finds 59 clusters of 286 extensions sharing temporal patterns in their fake reviews.

Turning to RQ2, we examine extensions with fake reviews to determine if they are also malicious. In total, we find 86 malicious extensions, with attacks ranging from stealing search query data from users to redirecting users to fake surveys to win prizes. These extensions share a total of over 64 million users. Although the number of downloads can also be manipulated [38], it is still staggering.

In addition to our manual analysis, we collaborate with Adblock Plus and Avast to demonstrate how FakeX can be leveraged to expand a seed list of malicious extensions. Using Adblock Plus' list of 18 newly discovered malicious extensions [36], we discover 16 new associated malicious extensions with millions of users, where attackers sometimes tried to improve malicious code. This practical deployment of FakeX resulted in public acknowledgments of our findings by Avast and Adblock Plus [35, 36] and the removal of all of these extensions from the Web Store by Google.

In summary, the paper's contributions are the following:

- We analyze reviews in the Web Store and identify four techniques for fake reviews (Section 2).
- Based on these techniques, we propose FakeX and describe our five novel methods to detect fake reviews (Section 3)
- We evaluate our methods on all reviews in the Web Store to demonstrate how FakeX detects fake reviews (Section 4).
- We show how clusters of extensions with fake reviews can be leveraged to find malicious extensions (Section 5).

We discuss limitations in Section 6, present related work in Section 7, and conclude the paper in Section 8.

We stress that although we uncover a correlation between fake reviews and malicious extensions, the techniques used by FakeX discover fake reviews rather than malicious extensions. Indeed, many of the extensions found with our method and highlighted in our analysis, whose review patterns indicate reputation manipulation, are not malicious as of February 2023 (see Appendix E).

Coordinated Disclosure, Ethical Considerations, and Open-source Artifacts. We have reported our findings to Google, including the malicious extensions we find with FakeX. In total, we find 86 and of these 44 have been removed so far.

In line with the ethical principles for cybersecurity research from the Menlo Report [41], our research does not cause any harm to users or developers. We include extension and user IDs and names to aid the reproducibility of the results presented in this paper. We open-source the code for FakeX[2].

2 FAKE REVIEWS ON CHROME WEB STORE

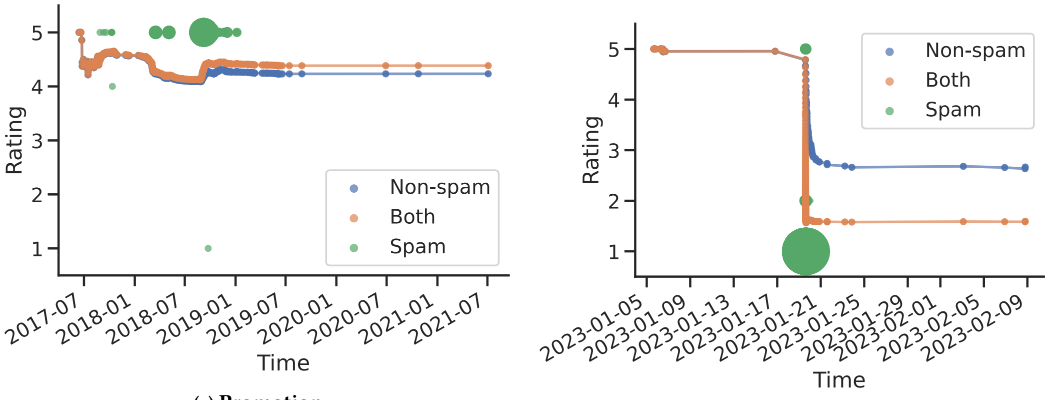
Reviews on Web Store. The Chrome Web Store [17] is the main repository for Chrome browser extensions. The Web Store allows users to write reviews and rate extensions. To submit a review, users must log in and install the extension. Users can leave a review, including text and a star rating, or rate the extension. These ratings, the number of stars given by users, are presented as an average across all user ratings in the Web Store. We cannot know the exact breakdown of the individual ratings nor when they were added. However, we can access the text, username, user ID, timestamp, and rating for each full review. Consequently, we can only know the entire rating history if all ratings come from reviews.

Reviews abused. Reviewers can influence how the Web Store ranks extensions. It is possible both to promote (posting positive reviews and high ratings) and demote (posting negative reviews and

[2] FakeX code and sample data: https://www.cse.chalmers.se/research/group/security/fakex

Table 1: Fake review techniques vs detection methods.

Fake Review Technique	Description	Detection Method(s)	Section
Disjoint Sets of Fake Accounts	Disjoint sets of multiple accounts for reviewing.	ATW, HVC	Sections 3.1 and 3.2
Fake Accounts	Using a set of accounts to review multiple extensions.	CoR	Section 3.3
Spamming	Large volume of reviews within a short period of time	Spam Detection	Section 3.4
Written Review Dominance	Only writing reviews but not rating	Written Ratio	Section 4.5



(a) Promotion

(b) Demotion

Figure 1: Promotion and demotion examples of browser extensions. "Spam" refers to reviews within three minutes of each other (More details in Section 3.4). The size of a point indicates the number of reviews. Every point represents new review activity of the extension. For "Non-spam" and "Both" the moving average up to that point is presented, as would be shown to real users of the Chrome Web Store.

low ratings) extensions [21]. Consequently, reviewers can damage extensions' reputation and reduce their perceived quality [14].

Fake review authors might cooperate and organize campaigns using different techniques to promote or demote extensions based on their reviews. For example, they might use multiple accounts or spread reviews out in time to avoid detection. These review campaigns aim to add as many reviews as possible in as short a time as possible, without being detected and removed [50, 51].

Figure 1a includes an example of reviewers promoting an extension[3] by artificially increasing the rating. We identify *spam* reviews as reviews within three minutes of each other (as we detail in Section 3.4). Around 2019, a larger spam campaign took place, after which we can see that the combined rating is higher than the non-spam one. We also include an example of reviewers demoting another extension[4]. In Figure 1b, we see a large amount of 1-star spam-marked reviews around the 21st.

Fake review techniques. We identify four techniques aimed at manipulating the reputation of extensions on the Web Store while evading detection. We refer to these reviews produced with the goal of reputation manipulation as *fake reviews.* The four techniques we focus on are *fake accounts*, *disjoint sets of fake accounts*, *spamming*, and *written ratio reviews.* Furthermore, we define a *review campaign* as a coordinated effort using multiple reviews to manipulate the reputation of one or more extensions. For example, if someone pays for 20 fake reviews to promote two extensions, these 20 reviews would be part of the same campaign.

1) *Disjoint Sets of Fake Accounts.* A motivated attacker might use disjoint—or partially disjoint, sets of fake accounts to write reviews. Using unique accounts makes fake reviewers less likely to be detected [31]. For example, five can review one extension from a set of ten accounts while the other five review another, making it harder to track the campaign.
2) *Fake Accounts.* A weaker version of the previous attack is to use multiple accounts, but not necessarily unique ones. Still, attackers might prefer to use a set of accounts instead of simply using one account to write all fake reviews to avoid detection. In Figure 2, we show how many extensions reviewers review. We observe that the majority of reviewers, over 91%, who review at least one extension only review one. That is, an overwhelming majority of reviewers only review one extension.
3) *Spamming.* Fake review authors may use automated tools or bots to submit many reviews without the need for human interaction. Unlike the previous methods, this approach requires analyzing the frequency and timing of reviews to distinguish them from legitimate user feedback.
4) *Written Review Dominance.* The Web Store allows users to rate extensions (1-5 stars) or write a review and rate the extensions.

[3] hgjdbeiflalimgifllheflljdconlbig
[4] fiikommddbeccaoicoejoniammnalkfa

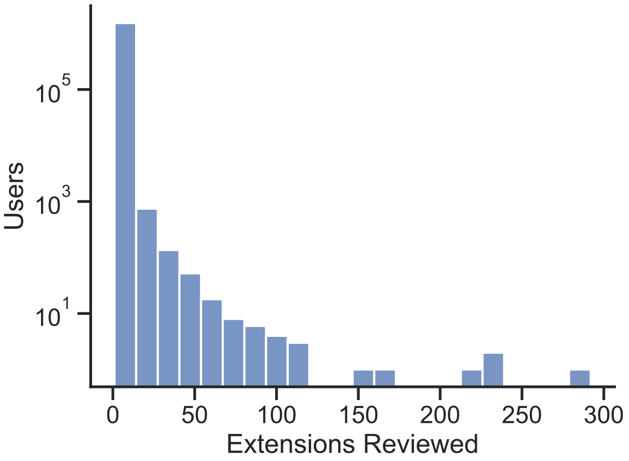


Figure 2: Distribution of the number of browser extensions users review, with at least one review, in the Web Store as of February 2023. On average these users review 1.16 extensions.

Since a valid account is needed for rating and reviewing, the attacker has no additional technical challenge. Adding text together with a rating is more persuasive. In the case of malicious extensions, malicious reviewers can include keywords such as "safe" and "secure" to trick users. Full-text reviews are also what is being provided by fake review services [15, 50–52].

Motivated by these manipulation techniques, we set out to propose a general framework for detecting fake reviews and evaluate it on reviews from the Web Store.

3 FAKEX: FRAMEWORK

This section presents FakeX, a framework that combines five methods to detect fake reviews of extensions in the Web Store. Our primary goal is to detect review campaigns (RQ1) and only then identify potentially malicious extensions (RQ2).

Detecting Fake Reviews in the Web Store. In the following, we present three methods to detect fake reviews of the Web Store: *Aggregated Time Window* (ATW), *Horizontal Vertical Clustering* (HVC), and *Co-Reviewer* (CoR) analysis. While all three methods attempt to detect abnormal review patterns by detecting coordinated reviews on extensions and generating clusters of extensions with shared behavior, they have different approaches. As we will see, the main difference between these methods is that CoR primarily targets clusters of reviewers reusing their accounts, while both ATW and HVC address the case where fake review authors create new accounts or multiple accounts are otherwise used.

3.1 Aggregated Time Window (ATW)

Intuition. The ATW method aims to link reviews posted within close temporal proximity. This is known in the literature as a *burst* and is formally defined as short periods of intensive activity followed by long periods of inactivity [3]. By focusing on the temporal aspect, instead of reviewer IDs or relationships, we can detect attacks using disjoint sets of accounts, as explained in Section 2.

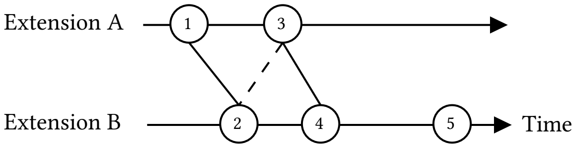


Figure 3: Example of two extensions with reviews and the connections ATW makes. The dotted line shows non-maximizing connections that discrete optimization will remove.

Figure 3 presents an example with two extensions, A and B, with circles representing the reviews they get over time. ATW will connect reviews one-to-one within the same time bursts. This effectively creates a graph with reviews as nodes and edges connecting them if these reviews are within the same burst. Multiple burst connections are possible. Review 2 can connect to either review 1 or 3 in this example. We maximize the number of connections. Note that if review 2 connects with review 3, then 1 and 4 will not be connected. ATW does not consider internal connections such as 1 to 3. Finally, we consider the individual and common (shared) number of reviews before clustering and filtering those that do not meet a specified threshold. In this example, reviews 1, 2, 3, and 4 are in common, while review 5 is not. This filter is crucial to avoid clustering all extensions with very frequently reviewed extensions.

Method. First, we connect all reviews in the same burst. At this stage, we do not have a one-to-one constraint. Second, we filter these connections with a threshold for the ratio of burst shared between two extensions to the max of the two extensions' total reviews. We remove any connection where the extensions share less than four bursts to further remove noise. This helps remove coincidental connections where one extension with few reviews happens to be paired up with a frequently reviewed extension with potentially hundreds of reviews. Given the constructed graph, we face the issue of overlapping connections, as the algorithm connects every review to every other in close temporal proximity. On this graph, we apply discrete optimization to match every review with at most one other, optimizing the total number of connections.

When implementing this algorithm, a critical detail is that no review is connected to another review of the same extension. The lack of such connections naturally makes the graph of one extension pair bipartite. Bipartite graphs are graphs in which vertices can be divided into two separate, non-intersecting groups. If a graph is bipartite, it also implies that the incidence matrix of that graph is guaranteed to be unimodular [30]. This property allows discrete optimization to be applied with low computational overhead.

Finally, after the discrete optimization ensures the reviews are connected one-to-one, we perform a second filtering. Similar to the first filtering, we ensure that the connected reviews make up a significant portion of the total reviews.

3.2 Horizontal Vertical Clustering: A Machine Learning Approach

Inspired by previous research comparing timeseries [39], we implement a Machine Learning (ML)-based approach. This approach involves clustering reviews into what we call "horizontal clusters" for reviews in the same time series of one extension and "vertical clusters" for clustering multiple time series/extensions using the centroids of the horizontal clusters produced before. Intuitively, a horizontal cluster represents a burst of review activity for one extension, and a vertical cluster represents a shared burst for multiple extensions. After this core idea, we denote our method, Horizontal Vertical Clustering (HVC). Similar to ATW, the focus is on temporal data, allowing us to detect attacks using disjoint sets of accounts, as explained in Section 2.

Specifically, we use the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm for horizontal and vertical clustering. DBSCAN [5] is an unsupervised clustering algorithm that groups based on an epsilon hyperparameter, which is the local radius for expanding clusters. In this method, epsilon is the threshold for the maximum time distance between reviews, or centroid of review clusters, within a group. This group consists of either reviews for a single extension or centroids of multiple extensions. It is worth mentioning that DBSCAN has been previously used in malware analysis [26] and clustering browser extensions for malware detection [37].

While DBSCAN performs well in its role as the clustering algorithm for this method, we are not taking advantage of some of its unique characteristics, such as finding arbitrarily-shaped clusters. Therefore, we believe that it can be substituted for other algorithms.

In Table 2, we include a real example illustrating how we use HVC to form clusters of extension reviews horizontally and vertically. In this example, we consider a vertical cluster comprising of three extensions A[5], B[6], and C[7]. We first create the horizontal clusters, i.e., grouping reviews written close enough in time per extension. This forms a summary of the review activity for a single extension. For instance, we can see that HVC clusters all the reviews within a timestamp of over 30 minutes (i.e., 9:46:42 and 10:40:36) of the C[7] extension in the same horizontal cluster (see "Horizontal Cluster" column of Table 2).

After that, we compute the centroid (Horizontal Centroid in the table), which serves as the datetime value for clustering extensions vertically (see Vertical Cluster column). Interestingly, in this example, the centroids of the reviews for these three extensions are within a radius of less than two minutes (0:01:32.3). We also include a graphical representation of the same example in Figure 4.

Table 2: Example of a vertical cluster DBSCAN produces together with its horizontal clusters. We use 0.0001 and 1.5e-06 as the epsilons for the horizontal and the vertical clusters, respectively. All the reviews of this table were written on the same day (2023-01-11) between 09:46:42 and 10:44:15.

Extension	Username	Time	Centroid
C[7]	Dennis	09:46:42	10:11:33
	Yuriy	09:57:52	10:11:33
	William	10:08:38	10:11:33
	Аркадий	10:23:55	10:11:33
	Jamie	10:40:36	10:11:33
B[6]	Dennis	09:48:46	10:13:19
	Yuriy	09:59:13	10:13:19
	William	10:10:18	10:13:19
	Аркадий	10:25:16	10:13:19
	Jamie	10:43:03	10:13:19
A[5]	Dennis	09:49:59	10:14:38
	Yuriy	10:00:31	10:14:38
	William	10:11:53	10:14:38
	Аркадий	10:26:32	10:14:38
	Jamie	10:44:15	10:14:38

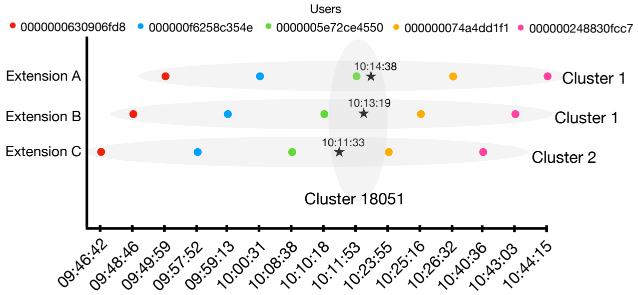


Figure 4: Example of five reviewers of three real extensions A[5], B[6], and C[7] that DBSCAN groups in the same vertical cluster (18051) using 0.0001 and 1.5e-06 as the epsilons for horizontal and vertical clustering. We mark the centroid of every horizontal cluster with a star.

3.3 Co-Reviewer

This method identifies connections between accounts that frequently review the same extensions, regardless of when this shared activity occurs in bursts. This approach helps uncover clusters of fake accounts, a technique we discuss in Section 2, which manipulates the reputation of a common set of extensions. The primarily targeted model of reputation manipulation for this method is *reviews campaigns*, detected when accounts are reused and review several heavily overlapping extensions.

In contrast to ATW or HVC, we preprocess the data here to filter out all accounts with only one written review. Since many reviewers only write one review (see Figure 2), we also improve runtime performance by removing them early in the process.

After this preprocessing, the algorithm iterates through each account and the extensions they reviewed, calculating the overlap between the current account and other accounts that reviewed the same extensions. This process establishes the degree of overlap between accounts. We use a threshold to form clusters of accounts with a high degree of overlap. Based on these clusters of accounts, we finally extract the list of shared extensions they reviewed.

[5] geokkpbkfpghbjdgbganjkgfhaafmhbo
[6] mpiihicgfapopgaahidedijlddefkedc
[7] lgjdgmdbfhobkdbcjnpnlmhnplnidkkp

Burst (min)	Clusters	Extensions
1	14	29
2	35	81
3	47	124
4	55	156
5	69	189
10	85	243
15	92	282
20	115	329
30	133	387
45	154	458
60	176	520

Table 3: Number of ATW clusters and included extensions for different bursts.

3.4 Spam Detection

Spam detection aims to find attackers who post as many reviews as possible in a very short time. For example, an extension might get a legitimate review once a day but then get ten reviews in just three minutes. Our spam detection method aims to detect this type of attack. To achieve this task, we define the time between two consecutive reviews in an extension as Δt. Then, for every pair of consecutive reviews, we mark them as suspicious if the Δt is less than a threshold, which we set to three minutes in this study. In Appendix C, we look closer at the general distribution of Δt for all extensions and further motivate our choice of threshold.

3.5 Written Ratio

This method leverages users' choice to leave a rating or attach text to their review. Since it takes extra effort to write text, we suppose that not all users who leave a rating will also write a review. We detect some abnormal review patterns by analyzing the ratio between the written reviews and ratings. An example of this type of attack is on the extension "D365-UI-Test-Designer"[8]. This extension has 141 ratings, all of which include written reviews, resulting in a written ratio of 100%. As such, this method will report extensions with a suspiciously high fraction of written reviews.

4 EVALUATION

We implement FakeX in Python and deploy the framework on a Windows computer using an AMD Ryzen 5 5600X CPU and 32GB of RAM. In this section, we present the results of FakeX together with our analysis and insights.

We crawled the Web Store as of February 9th, 2023. In total, we collected the extension's name, ID, and all written reviews of 1,782,702 reviews across 55,107 extensions (out of a total of 115,124 extensions). For every review of an extension, we have associated metadata: user's name, user's ID, review text, rating, timestamp of the initial review, and timestamp of the latest modification.

Table 4: Visualization of extensions with temporally shared reviews. The second and third columns represent the time and author of each review of Ninja Cut Unblocked, and the same goes for columns four and five, respectively, for X-Trial Racing Unblocked.

Date	Ninja Cut Unblocked		X-Trial Racing Unblocked		Delta
	Time	Reviewer	Time	Reviewer	
05/01	15:47:21	Caden	15:49:37	Patricia	2m 16s
16/01	10:32:32	Tracey	10:33:46	Monika	1m 14s
17/01	15:53:06	Lea	15:54:21	Ahsan	1m 15s
23/01	15:23:44	Hobart	15:24:36	Hobart	52s
24/01	19:02:13	Claire	19:03:34	Bernadette	1m 21s
02/02	17:35:35	Mason	17:36:58	Mason	1m 23s
07/02	15:14:24	Aroni	15:15:36	Aroni	1m 12s

4.1 Aggregated Time Window

The ATW method uncovers 59 clusters with three or more extensions, with an exceptionally high number of temporally shared reviews. For this evaluation, we use a burst length of 60 minutes. In Table 3, we present the number of clusters, including small clusters with only two extensions and extensions for different burst lengths. As we allow for a larger burst length, reviews farther apart in time can be clustered, and as such, the number of clusters increases. Using an excessive burst length will result in less precise results, including false positives.

To help visualize the context of temporally shared reviews, Table 4 shows the timestamp and the username from each review of two separate extensions[9]. Note that while some shared reviewers exist in this example, ATW would still cluster the extensions even if the reviewers were entirely different. In this example, the two extensions share all their reviews temporally, with less than 3 minutes between every correlated review.

To better understand the impact of burst length on the number and size of ATW clusters, we plot the sizes of clusters for different burst lengths in Figure 5. The highest density of clusters is always around 2-4 included extensions regardless of burst length. However, as the burst length increases, the largest clusters increase, as expected. This is shown in the figure by the increasing number of cluster size outliers. Increasing the burst period naturally lowers accuracy, though this is not a problem in the shown bursts. We can see in Appendix C that the review density for extensions in this dataset should, on average, be far more than our max burst length of one hour. Because of the low density of reviews, i.e., long time between reviews on average, the probability that reviews are written in short succession across multiple extensions is very low.

In Appendix B, we include two examples of real extensions that ATW detects and clusters together.

4.2 Horizontal Vertical Clustering

The HVC method, configured with a horizontal epsilon of 0.005 and a vertical epsilon of 5e-06, generates 69,618 vertical clusters. As mentioned in Section 3.2, these vertical clusters can be interpreted

[8] lfcoehhlodiaehjepemaogbgadfoipog

[9] iehblepfbknonbbncbidbaggofaomjop, ebllbagoalbkholngmhdlbcgfjhapdpk

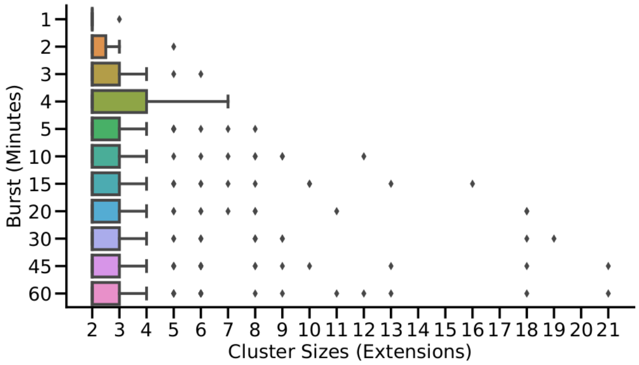


Figure 5: Cluster size (X-axis) of ATW instances based on different burst thresholds (Y-axis). Cluster size is in terms of extensions included.

as a shared burst of review activity for multiple extensions. This shared burst could be part, one execution, of a larger review campaign. Therefore, these clusters can be repeated—some extensions will be repeatedly grouped together.

Of these 69,618 vertical clusters, 30,344 are unique (not repeated). Of the 55,107 extensions, 11,226 (20.4%) are in these clusters. The unique clusters have an average membership of 2.09 extensions, with 6,241 clusters including more than two extensions. Of those clusters with more than two extensions, they have an average membership of 3.44.

Manually analyzing these 30,444 unique clusters, or even only the 6,241 clusters with more than two extensions, is infeasible. However, knowledge of some extensions being repeatedly grouped together can inform which extensions we select for analysis from these clusters. One simple way to select extensions is to pick those that most frequently occur. This can be interpreted as these extensions being most frequently manipulated in review campaigns.

Another way, which we ended up using, is to select maximal clusters—those that are not subsets of other clusters. The intuition here is that this filters out the noise of popular extensions being included in the clusters which otherwise describe a review campaign's single execution. Both extension or cluster selection mechanisms yield similar results, with the maximal cluster selection having slightly better quality in our opinion. Selecting maximal clusters with lengths greater than 2 yields 5,585 extensions, comprising 10.1% of the extensions in the entire dataset.

Even with smaller time epsilons, some interesting patterns of relationships between reviews that indicate fake reviews are evident in these clusters. In particular, we see in Table 5 that clusters often rediscover the relationship of extensions sharing a common developer (we share the extension IDs in Appendix E). These relationships can be found even when the extensions have a different scale in the number of reviews or have many reviews. This ability is unique to HVC among our methods (see Section 6.2). At the same time, HVC is liable to false positives due to coincidental reviews of popular extensions—in this example, a popular grammar and spelling checker plugin, unlikely to have been part of the review campaign with some obscure extensions by the same developer, is included in a cluster.

Table 5: A selection of HVC clusters that re-discover relationships that indicate fake reviews, namely a shared developer. These fake reviews are correlated even when extensions have many reviews and different scales of the number of reviews. With horizontal and vertical epsilons of 1e-06 and 5e-06.

Cluster	Extension name	Dev	Reviews
1	Grammar and Spelling checker…	Ginger	667
1	YT Thumbnail Downloader	Sagor	26
1	Ultimate Auto History Cleaner	Sagor	20
2	Share Google Contacts with…	GAPPS	84
2	Share Google Contacts Plugin	GAPPS	48
3	Aliexpress Search by image	ganes	227
3	Aliexpress Seller Check	ganes	183
4	SelectorsHub	Sanjay	903
4	SelectorsHub Pro	Sanjay	5
5	SelectorsHub	Sanjay	903
5	TestCase Studio	Sanjay	87

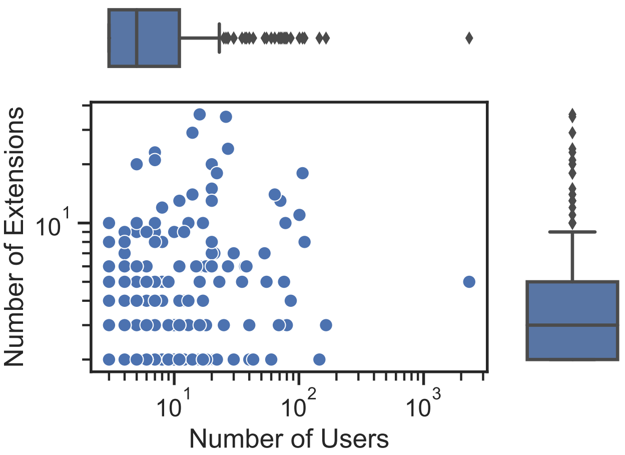


Figure 6: Clusters generated by ATW. Every point represents a cluster of a given number of extensions and reviewers.

4.3 Co-Reviewer Analysis

The Co-Reviewer analysis results in a total of 275 clusters. As we see in Table 12 (see Appendix A), only 9% of reviewers post more than one review, making the natural occurrence of these larger clusters of co-reviewing accounts uncommon.

This method produces clusters of extensions and reviewers similar to ATW. We can see the clusters generated by this method in Figure 6. The graph shows that the average cluster size is still very low, like the ATW results, but there are substantially more instances of large clusters. There are also generally more reviewers and extensions in the clusters of CoR. We hypothesize this is due to it being a more common attack that is also easier to detect. In the

Table 6: High scoring CoR cluster with 76 reviewers and five extensions. The table also includes the number of reviewers from the cluster that reviewed the extension and the ratio of all reviewers included in the cluster.

Extension name	Reviews from cluster	Ratio
Search by Image on Aliexpress	73	96.05%
Just vpn	71	93.42%
Search by Image on Alibaba	70	92.11%
Product search by image	69	90.79%
Boomtubes	35	46.05%

Table 7: Reviews detected as spam when executed with a threshold of three minutes. The ratio column shows the ratio between spam reviews and total reviews.

Rank	Name	Spam Reviews	Ratio	Rating
1	Ethos Sui Wallet	10,250	80%	5.0
2	Sui Wallet	4,095	79%	5.0
3	Swash	3,790	76%	4.8
4	Price Tracker...	3,752	68%	4.7
5	Glass wallet ...	3,713	55%	5.0
6	Fewcha Move Wallet	2,491	49%	5.0
7	Adobe Acrobat: PDF ...	2,317	14%	4.3
8	FlipShope - Price Tracker ...	1,961	45%	4.6
9	Morphis Wallet	1,832	75%	5.0
10	Bitfinity Wallet	1,672	74%	4.9

visualization, we also include the number of reviewers in clusters—notice that there are many reviewers in many clusters, emphasizing that this is a widely used attack technique. There is one extreme case of the massive outlier cluster in terms of included reviewers, that is a cluster containing 2,322 reviewers. This case results from these 2,322 reviewers mainly co-reviewing three extensions relating to the "Sui" cryptocurrency.

As expected, CoR analysis results reveal some overlap with the ATW method, as coordinated reputation manipulation on the same accounts across extensions creates patterns that both CoR and ATW discover in their clusters. ATW clusters occur when reviewers manipulate reputation simultaneously, creating a temporal correlation between their accounts.

In Table 6, we present a high-ranking cluster with 76 reviewers and five extensions. One of the reviewers in this cluster is "Mark", who reviewed all of the top four extensions in the table in only three minutes, which we regard as extremely fast and suspicious. Interestingly, the fifth extension in the table, "Boomtubes", was also reviewed by Mark three weeks later and has a much lower ratio than the other extensions. This could indicate that this cluster is comprised of two separate review campaigns, using slightly different sets of reviewers.

4.4 Spam Detection

The spam detection method uncovers 86,894 reviews, about 4.9% of all reviews, which are within three minutes of each other. In Table 7, the top ten extensions containing spam reviews are shown, where the method is run with the threshold of three minutes, meaning that every time the extension receives a review within three minutes after the last review was submitted, the spam count is increased by one. These numbers are incredibly high, a view supported by the vast top density. Also, notice that in the top ten, there are almost exclusively crypto-related extensions and price trackers, besides Adobe Acrobat. Adobe has a high number of spam reviews but a lower ratio compared to the others. Still, it is quite high compared to other popular extensions on the Web Store, e.g., NordVPN[10] (1%), MetaMask[11] (0.5%), and Skype[12] (0.5%).

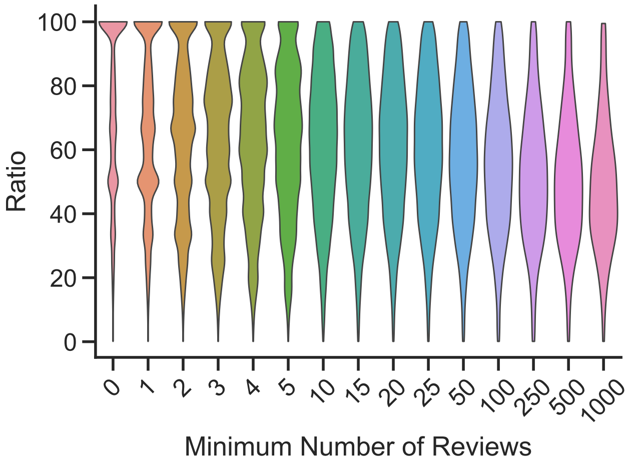


Figure 7: Relationship between thresholds (on number of reviews) and written ratios.

In Appendix D, we include a visual example of how spammed reviews look on the Web Store, including the timestamp of the reviews. We also highlight the rating distribution between spam reviews. The vast majority of spam reviews leave a rating of 5.

4.5 Written Ratio

Figure 7 depicts the distribution of written ratios, illustrating how unlikely extensions are to have these high ratios, especially considering how many reviews they have. The x-axis indicates the number of reviews the extension has to have strictly above to be included in the subset. For example, the first distribution includes all extensions. A big spike at the top indicates that many extensions have close to a 100% ratio. This is mainly due to extensions with few reviews, explaining the spikes at 33%, 50%, 66%, and 100% when including all extensions. However, as in the other distributions, the distribution quickly moves to the bottom for extensions with more reviews. The data reveal that extensions with a 100% written review ratio are highly improbable to occur naturally, especially in the subsets of extensions with above 25 reviews.

Table 8 presents the total ratings, written reviews, and written ratio for the top 10 scoring extensions. Every single one of the selected extensions has a written ratio of 100%. The "Percentile" column shows at what percentile the specific extensions rank if we

[10] fjoaledfpmneenckfbpdfhkmimnjocfa
[11] nkbihfbeogaeaoehlefnkodbefgpgknn
[12] lifbcibllhkdhoafpjfnlhfpfgnpldfl

Table 8: Top 10 scoring extensions by the Written Ratio method with their total ratings, written ratio, and percentile.

Name	Ratings	Written Ratio	Percentile
Opened or Not - Free Email...	705	100%	100.00%
TwitterScan - Find NFT Gems...	384	100%	99.96%
D365-UI-Test-Designer	141	100%	99.96%
DigiNovo screen sharing for...	136	100%	99.94%
AliExpress Search By Image...	126	100%	99.93%
Cashback beruby	116	100%	99.91%
RippleHouse	103	100%	99.87%
Jetstream	103	100%	99.87%
BROSH for LinkedIn and Gmail	102	100%	99.87%
Marucast Desktop Capture	99	100%	99.86%

exclude all extensions with strictly fewer reviews and compare the written reviews between the remaining extensions. For example, the top row means that no extension exists with the same number of reviews or more with the same written ratio, 100%.

Many cases in Table 8 present other types of suspicious behavior or other indications of being fake reviews. For example, in both "D365-UI-Test-Designer" and "DigiNovo screen sharing for A1 shop" *all* reviews contain the exact same review text, "I like it!". Many of the other ones show other signs of fake reviews, for example, a large number of reviews in a short time period.

Consider the distribution data shown in Figure 7—while extensions with only a few reviews can, and do, have high written ratios, extensions with many reviews should not consist of solely written reviews. Given this distribution, the near-100% written reviews of the extensions with 100s of reviews in Table 8 should be exceptionally rare. Every single extension in Table 8 is at least in the top sub-one percent of their respective threshold in Figure 7.

5 MALICIOUS EXTENSIONS

In this section, we explore the relationships between extensions with fake reviews and their maliciousness. We both present qualitative examples and a more quantitative case-study of the clusters generated by ATW. While we would want to analyze the maliciousness of all extensions, generating this ground truth is prohibitively slow and labor intensive.

5.1 Security Analysis

To evaluate the correlation between fake reviews and malicious extensions we first need a ground truth of malicious and benign extensions. While there are a plethora of malicious actions extensions can perform, we limit our analysis to the following attacks.

1) *Query stealing* [9]. This attack steals users' search queries from popular engines, either by presenting a search bar in a new tab or injecting code into search engines. A common pattern is that the attacker's search form will lead to a third-party server, which in turn redirects the user to the real search engine.
2) *History stealing* [9]. This attack focuses on tracking every URL a user visits. For example, by injecting code that fetches data from a third-party server on every URL.
3) *Affiliate fraud* [25]. In this attack, attackers try to make money when users shop online. If a user buys something on, for example, Amazon.com, a malicious extension might use *cookie stuffing* to give the extension developer a commission on any purchase.
4) *Survey scams* [8]. Survey scams force or trick users into completing online surveys in order to use a service. The surveys usually collect personal data while tricking the user into paying for a "prize" and stealing their credit card number.

We choose these attacks because they were explored in previous works and are relatively straightforward to detect. More specific attacks, like stealing information from social media sites, are difficult to detect as they might require valid accounts.

We manually analyze extensions by inspecting their source code and executing them to ensure malicious behavior exists. This task is labor-intensive, averaging over 10 minutes per extension. Since we are analyzing clusters, in most cases, the first extension takes longer to review. We can then generate a code signature to identify other extensions exhibiting similar characteristics. In total, we manually analyzed 299 extensions for malicious behavior.

Table 9: ATW cluster containing "New Tab" extensions. A *connected review* is a review that happened within the same burst as another in the cluster.

Extension Name	Total Reviews	Connected Reviews
SimpleTab	11	11
TopTab	10	8
NWTab	10	7
Handy Tab	9	6
Summer Tab	10	6
Amazing Tab	10	6
ToDoTab	10	6
Charming Tab	11	6
AmTab	10	5

5.2 Case-study of ATW clusters

To better understand the relationship between fake reviews and maliciousness, we perform a security analysis of the 286 extensions in the 59 clusters found by ATW. We detect 12 suspicious clusters (ratio of malicious extensions above 80% in Table 10). After a manual analysis, we find a cluster composed of "New Tab" extensions (see Table 9), which are notoriously malicious and often involve query stealing [9]. This confirms that ATW detects malicious extensions.

In Table 10, we report on the clusters with three or more extensions that ATW finds using bursts of 60 minutes. Interestingly, we note that many clusters are either strictly benign or strictly malicious. This indicates that review campaigns for malicious extensions do not mix with review campaigns for benign extensions. Furthermore, this supports ATW's ability to find meaningful clusters of related extensions.

5.3 New Tab Clusters

We further explore the extensions marked by all methods to find new malicious ones. We focus on a particular breed of extensions known for malicious behavior, the "New Tab" extensions [9]. These hijack the browser's home tab, replacing it with an alternative that modifies its functionality and appearance. In many cases, they also

Table 10: Number of clusters and percentage of malicious extensions in the clusters. From ATW using 60-minute bursts.

	Maliciousness						#Total
	0%	0%-25%	25%-50%	50%-75%	75%-100%	100%	
#Clusters	39	2	1	4	2	11	59
#Extensions	180	17	3	19	22	45	286

maliciously steal the users' search queries by redirecting traffic to their servers before redirecting them back to a real search engine.

Currently, there are 111 extensions flagged by all our methods. Among them, 13 are classified as "New Tab" extensions and, consequently, subjected to a thorough manual inspection. Astoundingly, each of these 13 turns out to be a query stealer. This discovery led to a further exploration of the clusters these culprits were associated with, according to the ATW and CoR methods. Following an exhaustive analysis of these clusters, the tally of malicious query stealers swells to 26. Interestingly, these extensions are dispersed across four distinct clusters, with the largest cluster harboring 16 of the 26 extensions.

5.4 Large Malicious Cluster

We look closer at the largest ATW cluster, composed of 18 extensions, of which 17 are malicious. All the malicious extensions use the same attack, namely *malicious surveys*. These are web pages that look like genuine surveys but trick the user into expensive subscriptions or malicious file downloads. For extensions, they also act as "human validation" needed before exposing malicious behavior.

Using FakeX, we find the game extension "Bloons Tower Defense Unblocked"[13] that was flagged by all methods except *Spam Detection*. We manually verify that the extension presents users with a survey from `stallmobiles.com`, which is part of a malware list [44]. However, the extension uses one level of redirection by first loading the `http://gameunblocked.pl/bloonstdgame-newtab` web page, which redirects to `stallmobiles.com`, making static code analysis harder. Interestingly, the only extension in this cluster that is not malicious had the same code structure, but no URL.

5.5 Expanding from Known Malicious Extensions

Finally, we demonstrate that FakeX can be used to expand a list of known malicious extensions. We collaborate with Adblock Plus and Avast on this particular deployment of FakeX. Utilizing a list of newly discovered 18 popular yet malicious extensions from Adblock Plus [36], we compare it against the results from ATW and CoR.

Interestingly, at least one of our methods flagged 16 of the 18 extensions. On delving into the clusters in CoR and ATW, we find that the union of clusters having at least one of these 16 extensions comprises 40 extensions. We present this list to Adblock Plus for further analysis. Based on this, Adblock Plus finds and confirms by Avast that 16 more extensions contain similar malicious code [36]. Furthermore, 8 of these used an improved version of the malicious code, compatible with the new manifest v3 [35]. Adblock Plus publicly acknowledged [35, 36] our contribution to discovering the additional 16 malicious extensions, and Google subsequently removed all of these extensions from the Web Store.

Table 11: A cluster of malicious extensions found by CoR compared with a subset found by ATW.

Extension	ATW	Malicious
Film Links Now | Default Search		✓
Autumn Tab	✓	✓
Primary Tab	✓	✓
Tasks Area		✓
Black Tab	✓	✓
Age Calculator		✓

6 DISCUSSION

In this section, we compare our methods, exploring their implications and relevance in the broader context. Additionally, we address the limitations inherent in our study, providing transparency about potential constraints and avenues for future research.

6.1 ATW and CoR

The main difference between ATW and CoR is that CoR primarily detects accounts with many reviews. In contrast, ATW often detects new accounts with only one singular review and strong temporal connections within clusters of unique accounts.

Since both ATW and CoR create clusters, we want to explore if overlaps in these clusters can help us find new malicious extensions. Since we already generated the ground truth for the ATW clusters, we search for CoR clusters that overlap with the ATW clusters. We present one such cluster in Table 11. If only ATW were used, only three malicious extensions would be found. However, combining ATW and CoR allows us to find three new extensions, which manual analysis confirms to also be malicious. This shows the power of combining the methods to find more malicious extensions.

6.2 Comparing ATW and HVC

Since both ATW and HVC base their clustering on temporal data, a comparison of the two methods is valuable. The methods are not directly comparable as ATW combines reviews from multiple time windows while HVC only considers two time radii.

In Table 9, ATW detects a cluster of nine extensions with fake reviews. However, it does miss the "NiceTab StartPage" [14] extension that HVC finds in a cluster together with "Charming Tab" [15]. ATW misses this extension because compared to the other nine in the cluster, this one had multiple reviews before the shared review campaign. Since ATW only clusters extensions with a significant overlap in reviews, it is not included. In general, ATW has difficulty clustering extensions that already had many reviews before getting fake reviews in the course of a review campaign. Reducing the threshold for the needed overlap could decrease the number of false negatives and allow ATW to cluster all ten of these extensions.

[13] monljmeefnongjlfefogaoldojpchhpg

[14] dobmhnlkolhhklmcaodfefhejoonalni

[15] kbnpeiabjlfcakokkpbcgalbgiljoddf

However, it might also increase the false positive rate and possibly add unrelated extensions to this cluster.

On the other hand, HVC does not cluster the other nine as a separate cluster. This is because other popular extensions, like NordVPN [16], also got reviews within only 20 minutes of the other new tabs. A general pattern we note is that ATW is often more precise in its clustering, with the downside of occasionally missing some extensions HVC finds in its clusters.

6.2.1 Goals of ATW and HVC - FP vs FN. While we do not have labeled fake review data to train a model on, we can still compare these two methods for detecting fake reviews based on related metrics. One metric can be malware labeling performance—which extensions containing malware are flagged by each method?

An obstacle is the lack of labeled data. A relatively independent labeling is that used by the Chrome browser—extensions can be labeled "malware", which will affect their ability to be loaded into the browser. Conversely, Chrome also has a notion of "good" extensions. The Chrome browser can allow some extensions in ESB (Enhanced Safe Browsing), which can be interpreted as being "safe" extensions. Using these labels provided by Chrome, we label "true positives" as being on the malware list, and "true negatives" as being on the ESB allowlist. False positives by this metric are extensions flagged by ATW or HVC that are not included in the Chrome malware list. Similarly, false negatives are extensions flagged by ATW or HVC that are considered safe by Chrome—on the ESB allowlist.

Metrics based on this labeling are conservative—we can establish some floor on false positives and false negatives. These Chrome-provided labels still do not cover the entirety of our dataset of extensions with reviews. Furthermore, these metrics are not an accurate description of performance.

This is plotted in Figure 8. The trends of ATW and HVC illustrate our previous observation of ATW having more accurate clusters—it has low false positives. When scanning for malware, a low false-positive rate can be valuable. ATW provides this low false-positive solution, while HVC can be used for better recall.

6.3 Focus on metadata

In this work, we solely focus on *metadata* to detect fake reviews of browser extensions in the Chrome Web Store. While *content* can also be used to detect fake reviews (see Section 7), we argue that detection mechanisms reliant on content are more easily eluded. Content can be faked easily and cheaply — fake review authors can copy existing review text, or generate them with a variety of methods. Metadata can also be faked, including with the fake review techniques discussed in Section 2. However, timestamps and user relations are harder and more expensive to fake, in that obscuring these temporal and user connections requires more time and user profiles to conduct a review campaign.

This metadata is not unique to the Chrome Web Store, and the fake review detection techniques we propose should be applicable to other online marketplaces. However, the timestamps utilized by the ATW, HVC, and Spam Detection methods are not always readily available - we note that the Yelp [23] and Amazon [34] datasets only have coarse date precision for their review timestamps.

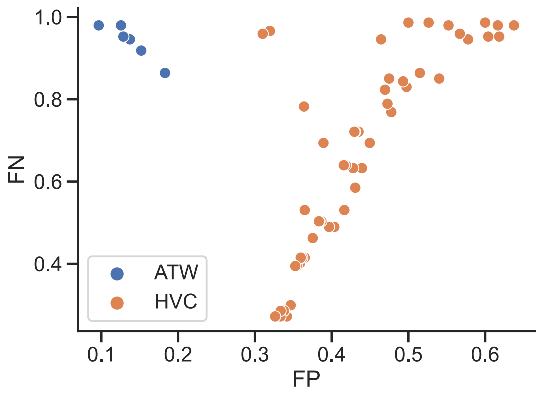


Figure 8: Performance of ATW and HVC according to Chrome malware labels, with each data point being a different parametrization of the respective methods. In general, ATW provides a low false-positive solution, while HVC offers low false-negatives.

7 RELATED WORK

Browser extensions. Researchers explore methods for detecting malicious extensions, ranging from scrutinizing downloads [38] and dynamically analyzing extension behavior, including information sent to external parties [9], to examining the sequences of API calls of common malicious actions [1]. Furthermore, studies have explored trends and values, detecting anomalous ratings [37], and monitoring extension executions to identify content modifications, such as injecting advertisements [2]. Static analysis to detect malicious JavaScript code could also be repurposed to detect malicious extensions [11]. Conversely, extensions have also served as vectors for attacks, i.e., exploiting some vulnerabilities in the extensions' source code, enabling activities like user tracking [47–49], acquiring sensitive information such as user history [6, 12], and facilitating remote code execution [12, 46]. Static analysis has been employed to discover such vulnerabilities [13, 53].

Pantelaios et al. [37] analyze anomalous extension ratings, code changes, and keyword pattern matching. However, their methods are limited to extensions with at least 50 reviews. FakeX focuses on the time component rather than the content of the reviews, and therefore, has no limitation on the number of reviews.

Despite the significant progress in browser extension security [9, 24, 25, 43], the orthogonal focus on user reviews and reputation manipulation in the Web Store, as highlighted in our research, represents a promising and fruitful direction that complements the prior studies.

Fake reviews. Fake review detection has been studied in other marketplaces, where users can also post reviews of products. However, prior methods in this area are often supervised and require ground truth labeling. This is not available for our application to extensions in the Web Store. Furthermore, the labeled datasets used in prior works can lead to biased results and methods. For example, datasets derived from Yelp business reviews are commonly used [4, 22, 33]. These methods assume that the user-submitted reviews Yelp does not label as "Trustful" are fake. This can introduce

[16] fjoaledfpmneenckfbpdfhkmimnjocfa

a bias towards learning the techniques Yelp uses internally for its filtering. Other approaches [10, 20] solicit fake reviews through Amazon Turk. This could also introduce a bias towards detecting fake reviews produced with a specific generating system. Unsupervised methods, such as FakeX, avoid being biased in this fashion.

Despite these difficulties with either applying previous fake-review detection approaches to the browser extension ecosystem, or evaluating their performance, these works are still useful to compare to the methods of FakeX. Fake review detection approaches can be grouped by the data used. Some methods use reviewer networks [22, 28, 40], similar to the CoR analysis in this paper. Methods using timestamps [29] have some similarities to ATW, HVC, and Spam detection in this paper. The Written Ratio method uses review text (in its presence/absence), similar to [20, 33]. Finally, some methods also combine multiple features [4, 10, 32] as FakeX does. Despite these surface similarities, FakeX offers a novel unsupervised framework for detecting fake reviews with methods that are suited to finding malware in browser extensions, primarily using temporal review graph features. We expand on this by comparing it to the mostly closely related works below.

Rathore et al. [40] obtain partial ground-truth information about fraud reviewer IDs by soliciting fake reviews for applications on the Google Play Store, using Fiverr, a platform that connects freelancers to people or businesses looking to hire. While a partial ground truth would be valuable to informing both our method and evaluation, soliciting fake reviews could lead to bias in the data. Furthermore, buying fake reviews will not provide useful temporal data, necessary for the central ATW and HVC methods of FakeX.

Li et al. [28] also use partial ground-truth, assuming fake review labels from the review-hosting site Dianping have high recall. This enables a Positive and Unlabeled (PU) learning approach, where the reviewer graph with 'fake reviewer' labels is iteratively extended from an initial set, using the association of shared IPs. The Web Store does not offer either fake review labels or user IP addresses.

He et al. [22] form a ground truth about Amazon product reviews by monitoring Facebook groups that act as marketplaces for buying and selling fake reviews. From these groups, they identify which products buy fake reviews and train a model on the review network to detect these. Given the absence of evidence linking concrete buyer-seller networks, such as Facebook groups, to fake reviews in the Web Store, investigating such a relationship emerges as a potential avenue for future research. Unlike the focus of this paper, FakeX employs CoR analysis to identify highly clustered fake reviewer networks, while ATW or HVC can uncover more nuanced relationships that exploit disjoint sets of fake accounts.

The method proposed by Liu et al. [29] is perhaps the closest to our application of ATW and HVC, in that they utilize review timestamp metadata to identify anomalous review activity. The authors crawl Amazon China for review records, then apply different time windows to bucket review activity for a particular product. The authors use an unsupervised clustering algorithm (isolation forest) to identify products whose review activity is anomalous. When applying this simple bucketing approach, we were unable to recreate the interesting fake-review and malware clusters produced by FakeX. A potential issue with applying their method to extension data is the larger timespan considered. Simple bucketing will yield a list of buckets from the dataset start time to end time, most of which will be empty. Comparing these high-dimensional points is challenging. In contrast, FakeX's HVC performs horizontal clustering to ease the somewhat analogous vertical clustering task.

Barbado et al. [4] propose a Fake Review Framework (F3), which combines reviewer and review text features, and apply this to the Yelp dataset. Reviewer features are mainly about the reviewer's activity, though this does include direct relationships such as friend networks. In contrast to this work, FakeX utilizes indirect relationships by reviewing the same extension through related reviews (ATW, HVC, Spam Detection, Written Ratio) or accounts (CoR).

Mukherjee et al. [32] also combine reviewer and review text features. They use expert labels of Amazon fake reviewer groups to hand-craft a set of spam indicator features containing reviewer behavior, relationships, and content similarity. The 'spamicity' of reviewers and groups is then iteratively refined with these features. This method does seem applicable to the Web Store setting, but the base features will likely have to be adjusted when transposed from the original setting. We did not re-implement this method to evaluate, as the source code is unavailable.

Other solutions are based on the links of the reviewers, reviews, products, and merchants [10], or the sentiment analysis of the reviews [20, 33]. FakeX does not use these additional features. While the knowledge graph proposed by Fang et al. [10] can contain the review graph features used in this paper, this method is inapplicable without both a more complete understanding of the fake review ecosystem for browser extensions, and labeled data to train models to both use the knowledge graph and evaluate its complex construction. The integration of additional features into the analysis of browser extensions is a promising avenue for future research.

8 CONCLUSION

We propose FakeX, a framework for detecting fake reviews in browser extensions. FakeX leverages five methods, ATW, HVC, CoR, Written Ratio, and Spam Detection, to identify extensions with fake reviews. Our evaluation unveils hundreds of review campaigns used on the Web Store, as well as, different attack techniques used in the campaigns. In particular, we find 59 clusters across 286 extensions with fake reviews sharing temporal patterns. This positively answers our first research question, whether reputation manipulation exists on the Web Store.

While fake reviews do not necessarily imply malicious intent, they put extensions with fake reviews into the spotlight and motivate further scrutiny for security risks. This leads to the positive answer to our second research question on leveraging our methods to detect malicious extensions. Using FakeX we find a total of 86 extensions with a total of 64 million users. After reporting to Google, 44 of these extensions were removed. Finally, we collaborate with Adblock Plus and Avast to demonstrate FakeX in action, expanding a seed list of newly detected malicious extensions to discover a further 16 malicious extensions with millions of users, where in some cases attackers tried to improve malicious code.

Acknowledgements. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, the Swedish Foundation for Strategic Research (SSF), and the Swedish Research Council (VR).

REFERENCES

[1] Anupama Aggarwal, Bimal Viswanath, Liang Zhang, Saravana Kumar, Ayush Shah, and Ponnurangam Kumaraguru. 2018. I Spy with My Little Eye: Analysis and Detection of Spying Browser Extensions. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE Computer Society, Washington, D.C., 47–61. https://doi.org/10.1109/EuroSP.2018.00012

[2] Sajjad Arshad, Amin Kharraz, and William Robertson. 2016. Identifying Extension-Based Ad Injection via Fine-Grained Web Content Provenance. In *Research in Attacks, Intrusions, and Defenses*, Fabian Monrose, Marc Dacier, Gregory Blanc, and Joaquin Garcia-Alfaro (Eds.). Springer International Publishing, Cham, 415–436.

[3] Albert-László Barabási. 2005. The origin of bursts and heavy tails in human dynamics. *Nature* 435, 7039 (2005), 207–211. https://doi.org/10.1038/nature03459

[4] Rodrigo Barbado, Oscar Araque, and Carlos A Iglesias. 2019. A framework for fake review detection in online consumer electronics retailers. *Information Processing & Management* 56, 4 (2019), 1234–1244.

[5] Derya Birant and Alp Kut. 2007. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering* 60, 1 (2007), 208–221.

[6] Quan Chen and Alexandros Kapravelos. 2018. Mystique: Uncovering Information Leakage from Browser Extensions. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. Association for Computing Machinery, New York, NY, USA, 1687–1700. https://doi.org/10.1145/3243734.3243823

[7] Chrome Extensions Stats. 2023. https://chrome-stats.com/t/extension

[8] Paul Ducklin. 2020. Anatomy of a survey scam – how innocent questions can rip you off. https://news.sophos.com/en-us/2020/06/22/anatomy-of-a-survey-scam-how-innocent-questions-can-rip-you-off/

[9] Benjamin Eriksson, Pablo Picazo-Sanchez, and Andrei Sabelfeld. 2022. Hardening the security analysis of browser extensions. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing* (Virtual Event) *(SAC '22)*. Association for Computing Machinery, New York, NY, USA, 1694–1703. https://doi.org/10.1145/3477314.3507098

[10] Youli Fang, Hong Wang, Lili Zhao, Fengping Yu, and Caiyu Wang. 2020. Dynamic knowledge graph based fake-review detection. *Applied Intelligence* 50 (2020), 4281–4295.

[11] Aurore Fass, Michael Backes, and Ben Stock. 2019. JStap: a static pre-filter for malicious JavaScript detection. In *Proceedings of the 35th Annual Computer Security Applications Conference* (San Juan, Puerto Rico, USA) *(ACSAC '19)*. Association for Computing Machinery, New York, NY, USA, 257–269. https://doi.org/10.1145/3359789.3359813

[12] Aurore Fass, Dolière Francis Somé, Michael Backes, and Ben Stock. 2021. DoubleX: Statically Detecting Vulnerable Data Flows in Browser Extensions at Scale. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, Republic of Korea) *(CCS '21)*. Association for Computing Machinery, New York, NY, USA, 1789–1804. https://doi.org/10.1145/3460120.3484745

[13] Aurore Fass, Dolière Francis Somé, Michael Backes, and Ben Stock. 2021. DoubleX: Statically Detecting Vulnerable Data Flows in Browser Extensions at Scale. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, Republic of Korea) *(CCS '21)*. Association for Computing Machinery, New York, NY, USA, 1789–1804. https://doi.org/10.1145/3460120.3484745

[14] FreeAddon. 2018. Warning: Fake one-star reviews & ratings are bombarding FreeAddon extensions! https://web.archive.org/web/20180929105259/https://freeaddon.com/fake-1-star-ratings-reviews-attack-by-hackers/

[15] GetReview. 2024. SEO Agency/Company London UK | Digital Marketing Agency in High Wycombe. https://getreview.co.uk.

[16] Review GG. 2022. Buy positive reviews online at cheap prices on review community. https://reviewgg.com/.

[17] Google. 2023. Chrome Web Store. https://chromewebstore.google.com/

[18] Google Developers. 2022. Spam and Abuse. https://developer.chrome.com/docs/webstore/program-policies/spam-and-abuse/.

[19] Google Developers. 2022. Spam policy FAQ. https://developer.chrome.com/docs/webstore/spam-faq/#ratings-and-reviews.

[20] Petr Hajek, Aliaksandr Barushka, and Michal Munk. 2020. Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. *Neural Computing and Applications* 32 (2020), 17259–17274.

[21] Mark A Harris, Robert Brookshire, and Amita Goyal Chin. 2016. Identifying factors influencing consumers' intent to install mobile applications. *International Journal of Information Management* 36, 3 (2016), 441–450.

[22] Sherry He, Brett Hollenbeck, Gijs Overgoor, Davide Proserpio, and Ali Tosyali. 2022. Detecting fake-review buyers using network structure: Direct evidence from Amazon. *Proceedings of the National Academy of Sciences* 119, 47 (2022), e2211932119.

[23] Yelp Inc. 2024. Yelp Open Dataset. https://www.yelp.com/dataset.

[24] Nav Jagpal, Eric Dingle, Jean-Philippe Gravel, Panayiotis Mavrommatis, Niels Provos, Moheeb Abu Rajab, and Kurt Thomas. 2015. Trends and Lessons from Three Years Fighting Malicious Extensions. In *24th USENIX Security Symposium (USENIX Security 15)*. USENIX Association, Washington, D.C., 579–593. https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/jagpal

[25] Alexandros Kapravelos, Chris Grier, Neha Chachra, Christopher Kruegel, Giovanni Vigna, and Vern Paxson. 2014. Hulk: Eliciting Malicious Behavior in Browser Extensions. In *23rd USENIX Security Symposium (USENIX Security 14)*. USENIX Association, San Diego, CA, 641–654. https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/kapravelos

[26] Joris Kinable and Orestis Kostakis. 2011. Malware classification based on call graph clustering. *Journal in computer virology* 7, 4 (2011), 233–245.

[27] Brian Krebs. 2021. Using Fake Reviews to Find Dangerous Extensions. https://krebsonsecurity.com/2021/05/using-fake-reviews-to-find-dangerous-extensions/

[28] Huayi Li, Zhiyuan Chen, Bing Liu, Xiaokai Wei, and Jidong Shao. 2014. Spotting Fake Reviews via Collective Positive-Unlabeled Learning. In *2014 IEEE International Conference on Data Mining* (Shenzhen, China). IEEE Computer Society, Washington, D.C., 899–904. https://doi.org/10.1109/ICDM.2014.47

[29] Wenqian Liu, Jingsha He, Song Han, Fangbo Cai, Zhenning Yang, and Nafei Zhu. 2019. A method for the detection of fake reviews based on temporal features of reviews and comments. *IEEE Engineering Management Review* 47, 4 (2019), 67–79.

[30] Jiří Matoušek and Bernd Gärtner. 2007. *Understanding and using linear programming*. Vol. 33. Springer, New York, NY, USA.

[31] Rami Mohawesh, Shuxiang Xu, Son N. Tran, Robert Ollington, Matthew Springer, Yaser Jararweh, and Sumbal Maqsood. 2021. Fake Reviews Detection: A Survey. *IEEE Access* 9 (2021), 65771–65802. https://doi.org/10.1109/ACCESS.2021.3075573

[32] Arjun Mukherjee, Bing Liu, and Natalie Glance. 2012. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web*. Association for Computing Machinery, Lyon, France, 191–200.

[33] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, Natalie Glance, et al. 2013. Fake review detection: Classification and analysis of real and pseudo reviews. *UIC-CS-03-2013. Technical Report* (2013).

[34] Jianmo Ni. 2018. Amazon Review Data (2018). https://nijianmo.github.io/amazon/.

[35] Wladimir Palant. 2023. How malicious extensions hide running arbitrary code. https://palant.info/2023/06/02/how-malicious-extensions-hide-running-arbitrary-code/.

[36] Wladimir Palant. 2023. More malicious extensions in Chrome Web Store. https://palant.info/2023/05/31/more-malicious-extensions-in-chrome-web-store/.

[37] Nikolaos Pantelaios, Nick Nikiforakis, and Alexandros Kapravelos. 2020. You've changed: Detecting malicious browser extensions through their update deltas. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*. Association for Computing Machinery, New York, NY, USA, 477–491.

[38] Pablo Picazo-Sanchez, Benjamin Eriksson, and Andrei Sabelfeld. 2022. No Signal Left to Chance: Driving Browser Extension Analysis by Download Patterns. In *Proceedings of the 38th Annual Computer Security Applications Conference* (Austin, TX, USA) *(ACSAC '22)*. Association for Computing Machinery, New York, NY, USA, 896–910. https://doi.org/10.1145/3564625.3567988

[39] Ivens Portugal, Paulo Alencar, and Donald Cowan. 2020. A Framework for Spatial-Temporal Trajectory Cluster Analysis Based on Dynamic Relationships. *IEEE Access* 8 (2020), 169775–169793. https://doi.org/10.1109/ACCESS.2020.3023376

[40] Punit Rathore, Jayesh Soni, Nagarajan Prabakar, Marimuthu Palaniswami, and Paolo Santi. 2021. Identifying groups of fake reviewers using a semisupervised approach. *IEEE Transactions on Computational Social Systems* 8, 6 (2021), 1369–1378.

[41] The Menlo Report. 2012. The Menlo Report. https://www.dhs.gov/sites/default/files/publications/CSD-MenloPrinciplesCORE-20120803_1.pdf.

[42] Get Reviews. 2020. Get Reviews. https://getreviews.buzz/.

[43] Florian Schaub, Aditya Marella, Pranshu Kalvani, Blase Ur, Chao Pan, Emily Forney, and Lorrie Faith Cranor. 2016. Watching them watching me: Browser extensions' impact on user privacy awareness and concern. In *NDSS workshop on usable security*, Vol. 10. The Internet Society, Reston, Virginia, U.S.

[44] ShadowWhisperer. 2023. Malware. https://github.com/ShadowWhisperer/BlockLists/blob/master/Lists/Malware.

[45] Rebecca Soares, Chrome Policy Benjamin Ackerman, and Anti-Abuse Team. 2020. Keeping spam off the chrome web store. https://blog.chromium.org/2020/04/keeping-spam-off-chrome-web-store.html

[46] Dolière Francis Somé. 2019. EmPoWeb: Empowering Web Applications with Browser Extensions. In *2019 IEEE Symposium on Security and Privacy (S&P)*. IEEE Computer Society, Washington, D.C., 227–245. https://doi.org/10.1109/SP.2019.00058

[47] Oleksii Starov, Pierre Laperdrix, Alexandros Kapravelos, and Nick Nikiforakis. 2019. Unnecessarily Identifiable: Quantifying the fingerprintability of browser extensions due to bloat. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 3244–3250. https://doi.org/10.1145/3308558.3313458

[48] Oleksii Starov and Nick Nikiforakis. 2017. Extended Tracking Powers: Measuring the Privacy Diffusion Enabled by Browser Extensions. In *Proceedings of the*

26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1481–1490. https://doi.org/10.1145/3038912.3052596

[49] Oleksii Starov and Nick Nikiforakis. 2017. XHOUND: Quantifying the Fingerprintability of Browser Extensions. In *2017 IEEE Symposium on Security and Privacy (S&P)*. IEEE Computer Society, Washington, D.C., 941–956. https://doi.org/10.1109/SP.2017.18

[50] Buy Review Store. 2024. Provide Online Reviews Marketing Five Star Rating & Reviews Services. https://buyreviewstore.com.

[51] Review Sub. 2022. Free or Buy Google Reviews, Amazon Reviews & more. https://www.reviewsub.com/.

[52] Buy Smm World. 2024. Buy Smm World - Digital Marketing And Reviews Service Provider. https://buysmmworld.com.

[53] Jianjia Yu, Song Li, Junmin Zhu, and Yinzhi Cao. 2023. CoCo: Efficient Browser Extension Vulnerability Detection via Coverage-guided, Concurrent Abstract Interpretation. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 2441–2455. https://doi.org/10.1145/3576915.3616584

A BROWSER EXTENSIONS REVIEWERS

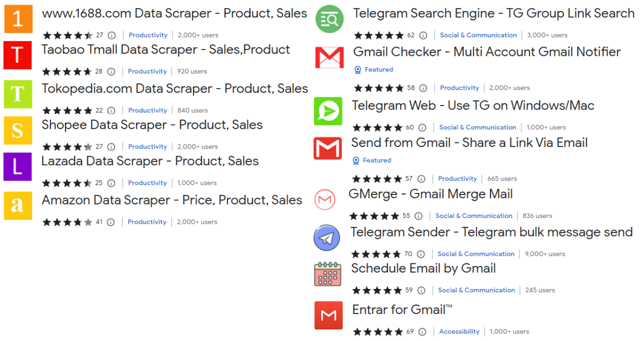
In Table 12, we include the overall number of extensions with reviews (first row), reviewers (second row), and reviews (third row). We further break down both the reviewers and the reviews they post by the type of reviewer. Reviewers are either Single, if they have posted only one review, or Multi, if they have posted multiple.

Table 12: Reviews distribution as of February 2023

	Reviewers		
Metric	Single	Multi	Total
Total Extensions			55,107
Total Reviewers	1 402 687 (91.29%)	133,819 (8.71%)	1 536 506
Total Reviews	1 402 687 (78.68%)	380,015 (21.32%)	1 782 702

B AGGREGATED TIME WINDOW EXAMPLES

Figure 9 depicts visual examples of high-scoring clusters. Notice the similarities in amount of reviews, which are closely related in quantity. They also share a temporal pattern of when the reviews were submitted, which is why ATW clustered them.



www.1688.com Data Scraper - Product, Sales
★★★★★ 27 | Productivity | 2,000+ users

Taobao Tmall Data Scraper - Sales,Product
★★★★★ 28 | Productivity | 920 users

Tokopedia.com Data Scraper - Product, Sales
★★★★★ 22 | Productivity | 840 users

Shopee Data Scraper - Product, Sales
★★★★★ 27 | Productivity | 2,000+ users

Lazada Data Scraper - Product, Sales
★★★★★ 25 | Productivity | 1,000+ users

Amazon Data Scraper - Price, Product, Sales
★★★★★ 41 | Productivity | 2,000+ users

Telegram Search Engine - TG Group Link Search
★★★★★ 62 | Social & Communication | 3,000+ users

Gmail Checker - Multi Account Gmail Notifier
Featured
★★★★★ 58 | Productivity | 2,000+ users

Telegram Web - Use TG on Windows/Mac
★★★★★ 60 | Social & Communication | 1,000+ users

Send from Gmail - Share a Link Via Email
Featured
★★★★★ 57 | Productivity | 665 users

GMerge - Gmail Merge Mail
★★★★★ 55 | Social & Communication | 836 users

Telegram Sender - Telegram bulk message send
★★★★★ 70 | Social & Communication | 9,000+ users

Schedule Email by Gmail
★★★★★ 59 | Social & Communication | 245 users

Entrar for Gmail™
★★★★★ 69 | Accessibility | 1,000+ users

Figure 9: Visual example of two clusters found by the ATW method.

Figure 10 compares reviews of two extensions from the Gmail cluster. Notice that they got reviews at the same time between extensions. In this case, it also happens to be the same reviewers, resulting in this particular pattern being clustered by CoR as well.

C REVIEW TIME DISTRIBUTION

In Figure 11 we see how reviews are distributed in time. Interestingly, the data seem to follow a log-normal distribution (the x-axis is in a logarithmic scale) where the average Δt is over 19.5 days. Also notice the spike at 1, which only contains reviews with a second or less time delta, which is the most extreme case of spam. We even observe 14 cases of sub-millisecond deltas; since the Web Store has only millisecond accuracy, these have the same recorded timestamp. Approximately, 95% of the reviews have a Δt of more than three minutes, making three minutes a reasonable threshold to find abnormally fast reviewing activity.

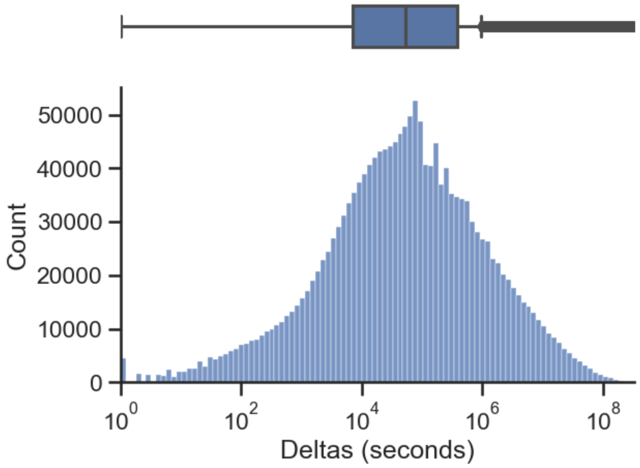
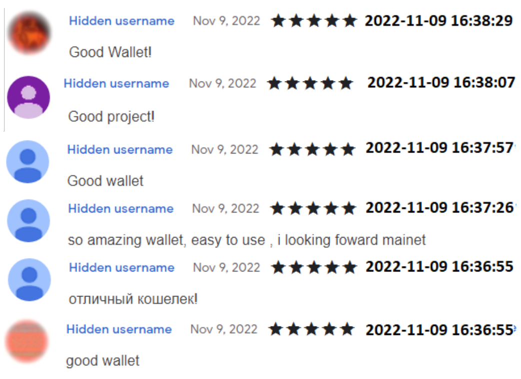


Figure 11: Data distribution of the difference in time between reviews (deltas) in an extension.

D SPAM DETECTION RATING

Figure 12 shows how spammed reviews look on the Web Store, including the timestamp of the reviews. Note that there are only a few seconds between each review, except for two reviews in the *same second.*



Hidden username Nov 9, 2022 ★★★★★ 2022-11-09 16:38:29
Good Wallet!

Hidden username Nov 9, 2022 ★★★★★ 2022-11-09 16:38:07
Good project!

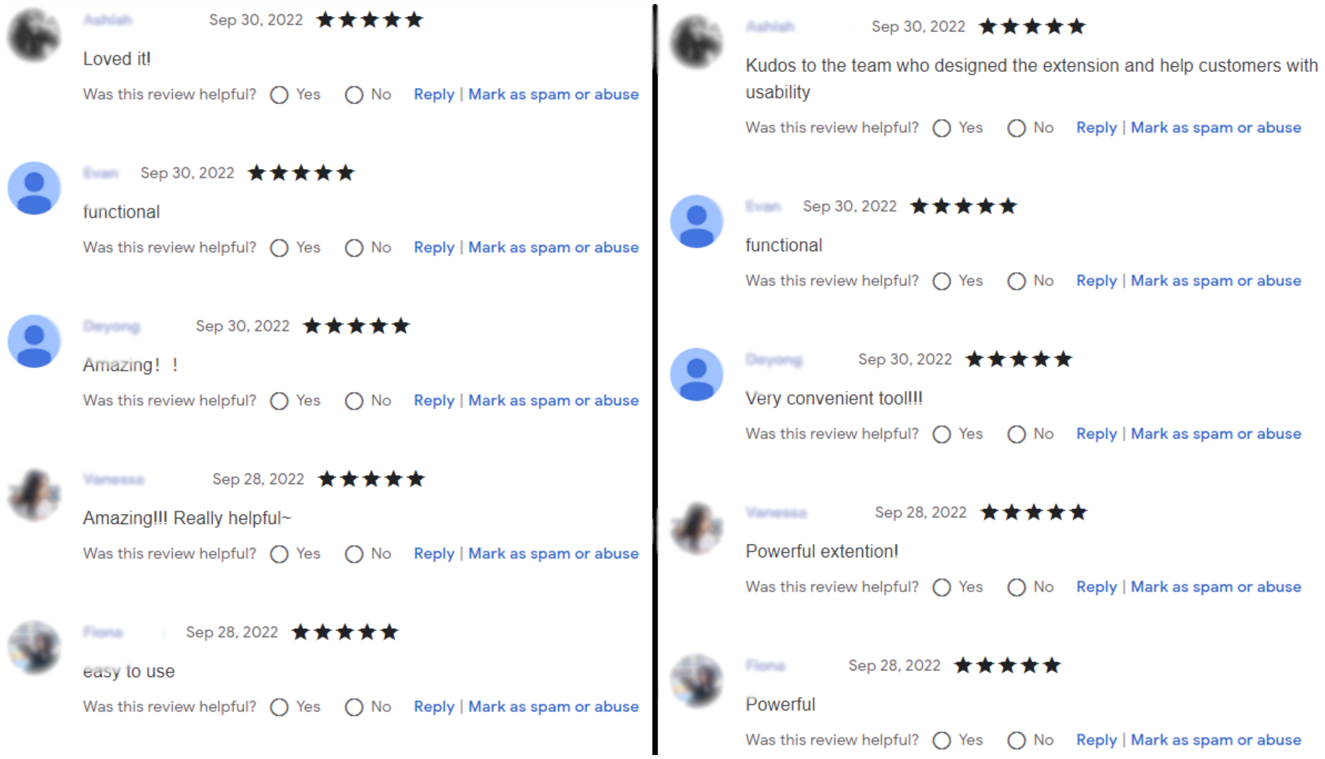
Hidden username Nov 9, 2022 ★★★★★ 2022-11-09 16:37:57
Good wallet

Hidden username Nov 9, 2022 ★★★★★ 2022-11-09 16:37:26
so amazing wallet, easy to use , i looking foward mainet

Hidden username Nov 9, 2022 ★★★★★ 2022-11-09 16:36:55
отличный кошелек!

Hidden username Nov 9, 2022 ★★★★★ 2022-11-09 16:36:55
good wallet

Figure 12: Spammed reviews with timestamps on the extension *Ethos Sui Wallet.*



Ashish Sep 30, 2022 ★★★★★

Loved it!

Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

Evan Sep 30, 2022 ★★★★★

functional

Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

Deyong Sep 30, 2022 ★★★★★

Amazing! !

Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

Vanessa Sep 28, 2022 ★★★★★

Amazing!!! Really helpful~

Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

Fiona Sep 28, 2022 ★★★★★

easy to use

Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

Ashish Sep 30, 2022 ★★★★★

Kudos to the team who designed the extension and help customers with usability

Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

Evan Sep 30, 2022 ★★★★★

functional

Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

Deyong Sep 30, 2022 ★★★★★

Very convenient tool!!!

Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

Vanessa Sep 28, 2022 ★★★★★

Powerful extention!

Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

Fiona Sep 28, 2022 ★★★★★

Powerful

Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

Figure 10: Temporally shared reviews between extensions, clustered by both ATW and CoR. Last names are removed from reviewers for ethical reasons.

Figure 13 presents the distribution of ratings within spam reviews, i.e., reviews made in less than three minutes of each other. The graph shows that a large majority of the spam reviews are five star reviews, indicating these reviews are mainly used to promote extensions.

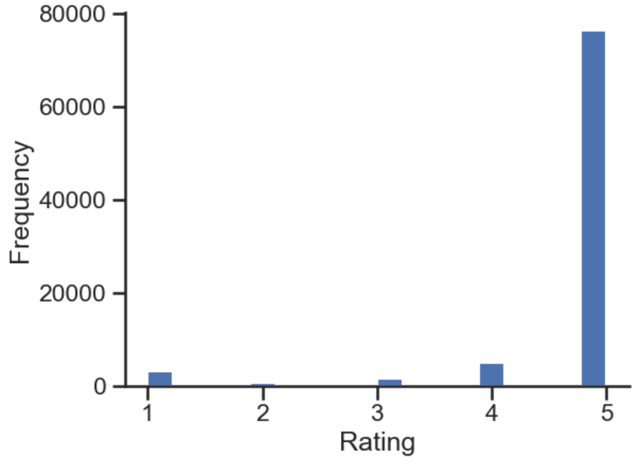


Figure 13: Rating distribution of spam marked reviews, using a threshold of three minutes.

E EXTENSION IDS FOR EXAMPLES NAMED IN TABLES

In Table 13, we present all the extensions used in tables throughout the paper together with their IDs, and if we consider them malware.

Table 13: Extension IDs for examples named in tables.

Table	Extension name	Extension ID	Malware
Table 5	YT Thumbnail Downloader	akfikgmhbajiaekdbgchbmhkceclncda	No
	Aliexpress Search by image	jkcacbjiofjgbnaknoojjboeiinempoa	No
	Grammar and Spelling checker by Ginger	kdfieneakcjfaiglcfcgkidlkmlijjnh	No
	SelectorsHub Pro	kodoloplfbnhlfcepehlafnbojbfgglb	No
	TestCase Studio	loopjjegnlccnhgfehekecpanpmielcj	No
	Ultimate Auto History Cleaner	nfnjemoofkhppjhjcehbddolbalmibkg	No
	Aliexpress Seller Check	mibmplgflabdmnnoncnedjfdpidjblnk	No
	SelectorsHub	ndgimibanhlabgdgjcpbbndiehljcpfh	No
	Share Google Contacts with Shared Contacts	nhmihkokjnmeaagjihlamgohjfmapehj	No
	Share Google Contacts Plugin	nllecbomigehlngfclbgjeghfmfajfgp	No
Table 6	Just vpn	apmomfapnjopaiiidbockbmbkklcfgni	Yes
	Search by Image on Aliexpress	chdmkeeecofpljchimdkliaknhaibkgm	Yes
	Boomtubes	igjenkfpfgfhoaagnmbbidjfbobmkohe	No
	Product search by image	inbbmabopknohmlmilkhjdidlmbhhofd	Yes
	Search by Image on Alibaba	pamfkmlimebecnfjoikmacloehbkhhoj	Yes
Table 7	Flipshope: Price Tracker and much more	adikhbfjdbjkhelbdnffogkobkekkkej	No
	Swash	cmndjbecilbocjfkibfbifhngkdmjgog	No
	Fewcha Move Wallet	ebfidpplhabeedpnhjnobghokpiioolj	No
	Adobe Acrobat: PDF edit, convert, sign tools	efaidnbmnnnibpcajpcglclefindmkaj	No
	Morphis Wallet	heefohaffomkkkphnlpohglngmbcclhi	No
	Price Tracker - Auto Buy, Price History	hegbjcdehgihjohghnmdpebepnoalode	No
	Bitfinity Wallet	jnldfbidonfeldmalbflbmlebbipcnle	No
	Glass wallet | Sui wallet	loinekcabhlmhjjbocijdoimmejangoa	No
	Ethos Sui Wallet	mcbigmjiafegjnnogedioegffbooigli	No
	Sui Wallet	opcgpfmipidbgpenhmajoajpbobppdil	No
Table 8	BROSH for LinkedIn and Gmail	bhjeblnbniahjoghbcngookdjdjjllde	No
	RippleHouse	dbjdhpndplhpppleinigdfnbibilkmod	No
	Opened or Not - Free Email Tracker	dmchdoholidpalbigibcgkkifklkcnil	No
	TwitterScan - Find NFT Gems & Trending Tokens	dmlbdfmbofhfnkneodciekpgaacbgdfo	No
	Jetstream	ijancdlmlahmfgcimhocmpibadokcdfc	No
	Marucast Desktop Capture	fjfnbddkahphhfhpmgknhgfbbnbbajkh	No
	D365-UI-Test-Designer	lfcoehhlodiaehjepemaogbgadfoipog	No
	AliExpress Search By Image | Rovalty	lijlkcihmpnnaijedioieaafmghjdnca	No
	Cashback beruby	lldknhffmfbndpbknmcckoelpidapidf	No
	DigiNovo screen sharing for A1 shop	pmpmejbonomjlbhphkkbeeeecpnknkpn	No
Table 9	NWTab	abcmjdhbopfnfkdonmkadfdghgipdeic	Yes
	Amazing Tab	agpoehmhgoieigdbjhgphpagmloehamn	Yes
	SimpleTab	ajjhojeehlipcemlodoncklkdoficgdi	Yes
	Summer Tab	dclbdlgnlaodfbjghpdjiodbnlicgalo	Yes
	AmTab	jalfhdofagnilegabknbiollkndbebei	Yes
	ToDoTab	jgealhbknfjhffedciigejkicpdnmhli	Yes
	Charming Tab	kbnpeiabjlfcakokkpbcgalbgiljoddf	Yes
	Handy Tab	kfnpaphhpnngikfmnofpkakbaekbafil	Yes
	TopTab	oilcbojeghcfkidelcmjbnbmaplfegbj	Yes
Table 11	Black Tab	coadpnfaiboiicgpgeggcpkkgpbbcele	Yes
	Age Calculator	gbaakcccffklmhhjhfamehdfcieojmbb	No
	Film Links Now | Default Search	hfgpkllpjcfpakbldligbhmkgkajjndk	No
	Primary Tab	mkakgkpinfpfapnliafpjkeccjphjgjf	Yes
	Autumn Tab	omcgiabgadgmpcplhdlniiddjbcocaah	Yes
	Tasks Area | Task Management Tool	pahcgdhpimolppohfdgcnfjeglelonab	No